J. Symbolic Computation (1994) **18**, 563–572

Term Rewriting for the Conjugacy Problem and the Braid Groups

JOHN PEDERSEN AND MARGARET YODER

Department of Mathematics, University of South Florida, Tampa, FL 33620-5700 USA

email jfp@math.usf.edu

(Received 11 October 1993)

We show how term rewriting can be applied to the conjugacy problem for finitely presented groups. The conjugacy problem is particularly important for the braid groups. We illustrate the term rewriting approach by applying it to the braid group B_3.

1. Introduction

Term rewriting has been applied to solve the word problem for many free and some finitely presented algebras (Evans, 1951; Knuth and Bendix, 1970; Pedersen, 1985) including some finitely presented groups (Le Chenadec, 1986; Benninghofen *et al.*, 1987) Dehn's two other fundamental algorithmic problems, the isomorphism problem and the conjugacy problem (Dehn, 1911) have not received as much attention from the term rewriting community, although see Book (1987) and the references there for some work on the conjugacy problem, using a different approach than here. In this paper we show how Knuth–Bendix style completion can be applied directly to solve the conjugacy problem. The conjugacy problem is particularly important for the braid groups, since it is only one step away from the holy grail of determining link isotopy via braid closures (Markov, 1945; Birman, 1974). We illustrate our approach by applying it to the braid group B_3. Although B_3 does not have some of the complexities of the higher groups, its word and conjugacy problems are still nontrivial. The braid groups have been the subject of considerable attention recently because of the new polynomial invariants for links (Jones, 1985, 1987; Turaev, 1988; Elrifai and Morton, 1990). These polynomial invariants are a "near miss" at a solution of the isotopy problem in that they provide reduced forms for isotopy classes but several isotopy classes may give the same reduced form. We feel it may be possible to extend the approach presented here to give a canonical form solution to the isotopy problem, where the canonical forms are obtained by term rewriting from braid representatives of links. In this case the invariants would be braid expressions rather than (Laurent) polynomials. Note that Freyd and Yetter's proof of the existence of the HOMFLYPT polynomial already uses a complete set of reductions to obtain the polynomial from a braid expression (Freyd *et al.*, 1985).

0747–7171/94/060563 + 10 $08.00/0

2. Term rewriting for the conjugacy problem

Let $G = \langle g_1, \ldots, g_n; R\rangle$ be a finitely presented group. For the rest of this paper, a *group word* is an expression $a_1^{\epsilon_1} \cdots a_k^{\epsilon_k}$, with $\epsilon_i \in \{-1, 1\}$, $a_i \in \{g_1, \ldots, g_n\}$ and $k \geq 0$. These are actually representatives of the congruence class of (an arbitrary parenthesization of) the indicated expression in the free group on the given generators. If $w = a_1^{\epsilon_1} \cdots a_k^{\epsilon_k}$ is a group word, then w^{-1} denotes the group word $a_k^{-\epsilon_k} \cdots a_1^{-\epsilon_1}$. Group words u and v are *conjugate* in G if there exists a group word w such that u and $w^{-1}vw$ are equal in G. The conjugacy problem for a given presentation is said to be solvable if there is an algorithm to determine of any given pair of words in the generators whether they are conjugate or not.

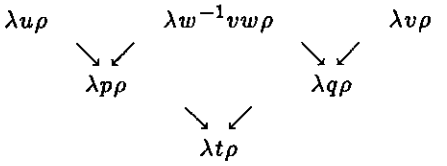
To treat the conjugacy problem by term rewriting, we express the group as a monoid in the standard way, introducing monoid generators corresponding to the inverse of each group generator. The usual completion process for a monoid term rewriting system uses the fact that if $u = v$ then $xuy = xvy$ for any x, y. If $=$ is to mean "conjugate" instead of equivalent, this is no longer true. The conjugacy relation applies to entire words, not subwords. Thus, the essential idea to enable completion to work for conjugacy is to make some reductions apply only to the whole string under consideration, not to any proper substring of it. This can be achieved by the use of left and right "end markers," which are introduced as two new generators, as demonstrated in Theorem 2.1 below. As usual, a string rewriting system is called canonical (or complete) if it is (locally) confluent and terminating. We assume the reader is familiar with basic results in term rewriting covering critical pairs and so forth—see Benninghofen *et al.* (1987) or Book (1982) for example.

THEOREM 2.1. *Let $G = \langle g_1, \ldots, g_n; S\rangle$ be a finitely presented group. Let λ and ρ be new symbols and let T be a canonical string rewriting system for the congruence induced on the free monoid on the generators $\{g_1, \ldots, g_n, g_1^{-1}, \ldots, g_n^{-1}, \lambda, \rho\}$ by*

$$S \cup \{g_i g_i^{-1} = e,\ g_i^{-1} g_i = e, \lambda g_i x g_i^{-1} \rho = \lambda x \rho,\ \lambda g_i^{-1} x g_i \rho = \lambda x \rho : i = 1, \ldots, n\}, \qquad (1)$$

where x is a variable quantifying over group words only. Then words u and v are conjugate in G if and only if $\lambda u \rho$ and $\lambda v \rho$ have identical reduced forms under T.

PROOF. Suppose u is conjugate to v. Then there exists a group word w such that u and $w^{-1}vw$ are equivalent in G. Since T is a canonical system equivalent to (1), which includes the defining relations S of G, both u and $w^{-1}vw$ will reduce to a common group word, say p, under T. Therefore, $\lambda u \rho$ and $\lambda w^{-1} v w \rho$ both reduce to $\lambda p \rho$ under T. Let $w = a_1^{\epsilon_1} \cdots a_k^{\epsilon_k}$ with $a_i \in \{g_1, \ldots, g_n\}$ and $\epsilon_i \in \{1, -1\}$. Then $\lambda w^{-1} v w \rho = \lambda a_k^{-\epsilon_k} \ldots a_1^{-\epsilon_1} v a_1^{\epsilon_1} \ldots a_k^{\epsilon_k} \rho$. This latter expression is equivalent under (1) to $\lambda v \rho$. Hence, under T, both $\lambda w^{-1} v w \rho$ and $\lambda v \rho$ reduce to a common form, say $\lambda q \rho$ (refer to the diagram below). We also have $\lambda w^{-1} v w \rho$ reducing to $\lambda p \rho$ from above. Thus, using the confluence of T again, $\lambda q \rho$ and $\lambda p \rho$ reduce under T to a common form, say $\lambda t \rho$. This is then a common reduced form of $\lambda u \rho$ and $\lambda v \rho$ under T.



For the converse, we first establish the
Claim. If $\lambda u \rho \to_T w$ then w has the form $\lambda v \rho$ for some v not containing λ or ρ, and v is conjugate to u in G.

Indeed, if $\lambda u \rho \to_T w$ then $\lambda u \rho =_{(1)} w$, since T is equivalent to (1). Thus, there is a sequence $w_0 = \lambda u \rho, w_1, w_2, \ldots, w_n = w$ of words w_i with each $w_i = w_{i+1}$ by a single substitution of a left side of a relation in (1) for the right side, or vice versa. But each relation in (1) preserves the properties of the claim, that is, they preserve the end markers λ and ρ and they preserve conjugacy of u and v in $\lambda u \rho$ and $\lambda v \rho$. This establishes the claim.

Now suppose $\lambda u \rho$ and $\lambda v \rho$ have identical reduced forms under T. Using the claim repeatedly, this common reduced form is some $\lambda w \rho$ where w is conjugate to u and to v. Conjugacy is an equivalence relation, and thus u is conjugate to v. QED.

Notice that, since the original relations S are included in the set to be completed, a complete set of reductions corresponding to S will be included in a complete set for (1). Thus, a prerequisite for using this method is being able to complete S. Also notice that the theorem does not automatically give a solution to the conjugacy problem in the group. A canonical set T can be enumerated by applying a Knuth–Bendix style completion procedure to S, but to determine algorithmically whether the words $\lambda u \rho$ and $\lambda v \rho$ have identical reduced forms under T, it must be decidable whether a given word contains an instance of a left side of a rule in T. We record this as

COROLLARY 2.2. *With G and T as in the theorem, if the set $\{w : w$ contains a left side of some rule in $T\}$ is recursive, then the conjugacy problem for G is solvable.*

To automate this approach to solving the conjugacy problem, a completion procedure can be applied to the set S, but some special provision must be made for the condition about subwords not containing the end markers. Existing completion programs do not appear to easily allow such a condition, but a special "conjugacy completion" program could clearly be developed. We have not yet undertaken this task, being content for the moment with hand calculations, such as the one for B_3 presented in section 4.

It may be important to note that if the approach were automated, then human work is still likely to be required, since it seems that complete sets for conjugacy will very often be infinite. After the program has generated some number of rules, a human can look over them with a view to finding patterns and possibly expressing them as finitely many rule schemata, conjectured to form a complete set. The same situation can occur for ordinary rewrite-completion (for the word problem) when a finite set is not produced by the completion program. Indeed, the authors have obtained a word problem solution for B_n in this way. In these cases, since the completion process has never terminated, it is of course necessary to supply a proof that the proposed schemata do constitute a complete rewriting system.

As noted above, a complete set for conjugacy contains a complete set for the original group relations, that is, a solution to the word problem for the group. In practice it is convenient to find this first, which can be done with existing completion programs. The word problem normal forms can then be examined, and completion for the extra conjugacy rules can be undertaken. We illustrate this approach next for the braid group B_3, starting with a term rewriting solution of its word problem. Of course, solutions to the word and conjugacy problems for the braid groups are known (Garside, 1969;

Jacquemard, 1990; Thurston, 1992), but we are not aware of any using term rewriting and we expect the term rewriting approach to give new insights into a combinatorial attack on the link isotopy problem.

3. The word problem for B_3

The standard presentations for Artin's braid groups (Artin, 1947) are

$$B_{n+1} = \langle \sigma_1, \ldots, \sigma_n;\ \sigma_i\sigma_j = \sigma_j\sigma_i, \quad |i-j| \geq 2,$$
$$\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}, \quad 1 \leq i \leq n-1 \rangle.$$

In particular, $B_3 = \langle \sigma_1, \sigma_2;\ \sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2 \rangle$. Thus B_3 is simpler than the other braid groups in that no commutation relations are involved. However, it still provides a nontrivial example of the theory.

To obtain a complete set of reductions for B_3 it is convenient to change to the generators $a = \sigma_1\sigma_2$ and $b = \sigma_1\sigma_2\sigma_1$. Note that by the original relations, we also have $b = \sigma_2\sigma_1\sigma_2$ and thus $\sigma_1 = a^{-1}b$ and $\sigma_2 = ba^{-1}$. The generator b represents a "half twist" of the braid strands and it figures prominently in most combinatorial work on braids – see (Birman,1974) for example. In terms of these generators, the group presentation for B_3 is $\langle a, b; a^3 = b^2 \rangle$, as the reader may verify (see also Kapur and Narendran,1985). For completion purposes, we introduce another (redundant) generator, $c = a^3$. Expressing the resulting group presentation as a monoid presentation in the usual way gives the following monoid presentation for B_3, in which the relations have been labelled (A)–(H) for later reference and e is used for the empty word (monoid identity element):

$$M_3 = \langle a, b, c, a^{-1}, b^{-1}, c^{-1};\ \begin{array}{ll} \text{(A) } aaa = c, & \text{(B) } bb = c, \\ \text{(C) } aa^{-1} = e, & \text{(D) } a^{-1}a = e, \\ \text{(E) } bb^{-1} = e, & \text{(F) } b^{-1}b = e, \\ \text{(G) } cc^{-1} = e, & \text{(H) } c^{-1}c = e \rangle. \end{array}$$

A finite complete set of rewrite rules for this presentation is provided in the next theorem.

THEOREM 3.1. *The set, T, of rewrite rules* (1)–(10) *below is a complete (confluent and terminating) string rewriting system for M_3, and thus provides a solution to the word problem for B_3.*

$$\begin{array}{ll} (1)\ a^{-1} \to c^{-1}aa, & (2)\ b^{-1} \to c^{-1}b, \\ (3)\ cc^{-1} \to e, & (4)\ c^{-1}c \to e, \\ (5)\ aaa \to c, & (6)\ bb \to c, \\ (7)\ ac \to ca, & (8)\ ac^{-1} \to c^{-1}a, \\ (9)\ bc \to cb, & (10)\ bc^{-1} \to c^{-1}b. \end{array}$$

PROOF. This set of rules can be obtained by applying rewrite-completion to the set (A)–(H), using a weighted dictionary order with weights of 1 for a, b, c, and c^{-1}, and 3 for a^{-1} and b^{-1} and ordering the generators as $c < c^{-1} < a < b < a^{-1} < b^{-1}$. Alternatively, one can show directly that T is locally confluent, terminating, and equivalent to the relations of M_3. Verification is left to the reader or the reader's favourite completion program.

We shall call a word in M_3 *T-reduced* if it does not contain any of the left sides of the relations in T. For later reference we note the form of T-reduced words, which, as can

be seen from (1)–(10), are all words with powers of c at the front followed by a string of a's and b's containing no block of aaa or bb. That is:

COROLLARY 3.2. *The T-reduced words are precisely all words of the form*

$$c^n L(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g R,$$

where n is any integer, $p_i, q_i > 0$, $v, g, f \geq 0$, (with the product taken to be e if $v = 0$), $L \in \{e, b\}$, and $R \in \{e, a, aa\}$, and products involving e are suitably interpreted.

4. The conjugacy problem in B_3

Let S be the set of relations in the monoid presentation M_3 of section 3. Since c is defined in terms of a and b, then according to Corollary 2.2, a term rewriting solution of the conjugacy problem for B_3 will be given by a suitable completion of the monoid presentation

$$C_3 = \langle a, b, c, a^{-1}, b^{-1}, c^{-1}, \lambda, \rho;\ Y\rangle,$$

where $Y = S \cup \{\lambda g^{\epsilon} Q g^{-\epsilon}\rho = \lambda Q \rho : g \in \{a, b\}, \epsilon = \pm 1$, and Q does not contain λ or ρ $\}$.

We will show that a complete set, Z, of rewrite rules for C_3 is given by the ten rules of T together with rules (11)–(15) below. In these rules, n is any integer but $v \geq 0$ and $f, g, p_i, q_i > 0$. Any word $\prod_{k=i}^{j} w_k$ with $i > j$ is, by definition, the empty word e. In rule (15) we need to choose a sequence from the set $\{p_j q_j p_{j+1} q_{j+1} \cdots p_v q_v p_1 q_1 \cdots p_{j-1} q_{j-1} : 1 \leq j \leq v\}$ of even-length cyclic permutations of a sequence of (pairs of) positive integers. We will choose the one that is first in lexicographic order. This sequence will be called the *smallest cycle* of $p_1 q_1 \ldots p_v q_v$. Each of the additional rules in Z is actually a schema, describing infinitely many rules. Nevertheless, we shall refer to them as single rules and the decidability condition of Corollary 2.2 is easily seen to hold. W denotes a nonnegative word in M_3. Rules (11)–(15) are:

(11) $\lambda c^n aW a\rho \to \lambda c^n aaW\rho$, $W \neq e$, aWa $T-$reduced,

(12) $\lambda c^n bW\rho \to \lambda c^n Wb\rho$, W contains a and does not contain c,

(13) $\lambda c^n (ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g\rho$
$\to \lambda c^n (aab)^g (ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})\rho$, $g > 0$, $f + v > 0$,

(14) $\lambda c^n (ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g\rho$
$\to \lambda c^n \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g (ab)^f\rho$, $f > 0$, $g + v > 0$,

(15) $\lambda c^n \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})\rho \to \lambda c^n \prod_{k=j}^{v}((aab)^{p_k}(ab)^{q_k}) \prod_{k=1}^{j-1}((aab)^{p_k}(ab)^{q_k})\rho$,
$v > 1$, $p_j q_j p_{j+1} q_{j+1} \cdots p_v q_v p_1 q_1 p_2 q_2 \cdots p_{j-1} q_{j-1}$ is the smallest cycle of $p_1 q_1 p_2 q_2 \cdots p_v q_v$, $j \neq 1$.

In the rest of this paper, $u \to_x v$ shall mean that v is obtained from u by 0 or more applications of x.

These rules were obtained by manually carrying out a process similar to that outlined in the discussion on automating conjugacy completion following Corollary 2.2. Starting with the rules given in Theorem 3.1, augmented for conjugacy as defined in Theorem 2.1, many critical pairs were calculated, and patterns in the resulting rules were examined, leading to the rule schemata given above. Note that, even if the "conjugacy completion" process were automated, a proof of completeness would still be necessary in this case because the rule set is infinite.

THEOREM 4.1. *The rules Z are a complete set of string rewriting rules for C_3 satisfying the condition of Corollary 2.2 and thus they solve the conjugacy problem for B_3.*

Before presenting the proof, which will occupy most of the rest of the paper, let us give an example of the theorem's use. Consider the braids $\sigma_1\sigma_2^{-1}\sigma_1^{-1}$ and σ_1^{-1}, illustrated in Figure 1.

Rewriting σ_1 as $a^{-1}b$ and σ_2 as ba^{-1}, we compute

$$\begin{aligned}\sigma_1\sigma_2^{-1}\sigma_1^{-1} &= a^{-1}bab^{-1}b^{-1}a \to_2 a^{-1}bac^{-1}bc^{-1}ba \to_1 c^{-1}aabac^{-1}bc^{-1}ba\\ &\to_{10} c^{-1}aabac^{-1}c^{-1}bba \to_6 c^{-1}aabac^{-1}c^{-1}ca \to_4 c^{-1}aabac^{-1}a\\ &\to_{8,10} c^{-1}c^{-1}aabaa \to_{11} c^{-1}c^{-1}aaaab \to_5 c^{-1}c^{-1}cab \to_4 c^{-1}ab\end{aligned}$$

and $\sigma_1^{-1} = b^{-1}a \to_2 c^{-1}ba \to_{11} c^{-1}ab$

The final forms given are identical under Z, so the theorem shows that these braids are conjugate in B_3. Indeed, $\sigma_2^{-1}\sigma_1^{-1}\sigma_2 = \sigma_1\sigma_2^{-1}\sigma_1^{-1}$.

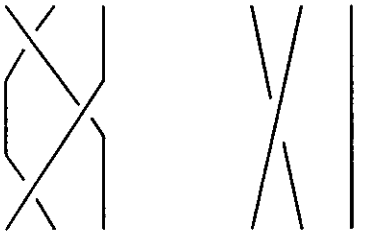
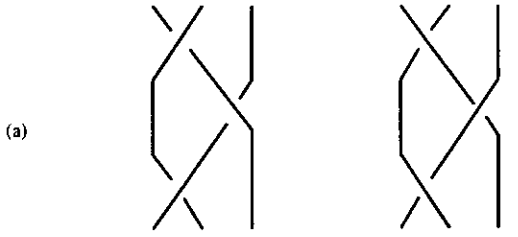


Figure 1. Two conjugate braids.

Now consider the braids $\sigma_1^{-1}\sigma_2\sigma_1^{-1}$ and $\sigma_1\sigma_2^{-1}\sigma_1$, illustrated in Figure 2(a).



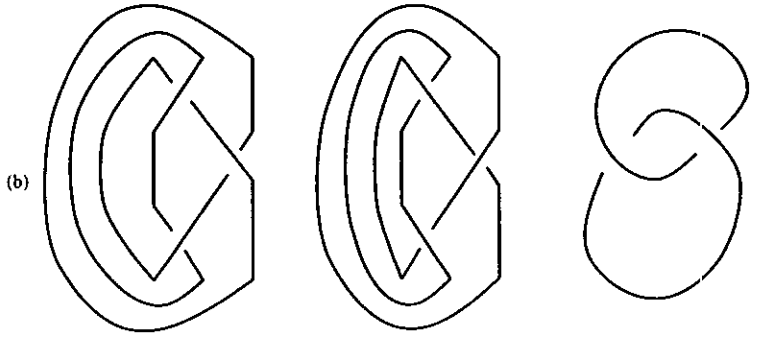


Figure 2. (a) Two non-conjugate braids.
(b) Their closures, which are both isotopic to the link at right.

Again rewriting σ_1 as $a^{-1}b$ and σ_2 as ba^{-1}, we compute

$$\begin{aligned}\sigma_1^{-1}\sigma_2\sigma_1^{-1} &= b^{-1}aba^{-1}b^{-1}a \to_{12} c^{-1}baba^{-1}c^{-1}ba \to_1 c^{-1}babc^{-1}aac^{-1}ba\\ &\to_{8,10} c^{-3}babaaba \to_{12} c^{-3}abaabab \to_{14} c^{-3}aababab\\ \sigma_1\sigma_2^{-1}\sigma_1 &= a^{-1}bab^{-1}a^{-1}b \to_1 c^{-1}aabab^{-1}c^{-1}aab \to_2 c^{-1}aabac^{-1}bc^{-1}aab\\ &\to_{8,10} c^{-3}aababaab \to_{13} c^{-3}aabaabab.\end{aligned}$$

and the final forms given are irreducible under Z, so the theorem implies that these braids are not conjugate in B_3. However, their closures are isotopic (define the "same" link), as shown in Figure 2(b).

PROOF. We prove local confluence, termination and equivalence to the rules in C_3. For local confluence, note that T is a subset of Z, and is complete in itself, by Theorem 3.1. The only overlaps to check are listed below, with their resolutions.

(12/6) $\lambda c^n bbW\rho \to_{12} \lambda c^n bWb\rho \to_{12} \lambda c^n Wbb\rho \to_6 \lambda c^n Wc\rho \to_{7,9} \lambda c^{n+1}W\rho$
$\to_6 \lambda c^{n+1}W\rho$

(12/T) If $W \to_T c^kW'$,
$\lambda c^n bW\rho \to_{12} \lambda c^n Wb\rho \to_T \lambda c^n c^k W'b\rho$
$\to_T \lambda c^n bc^k W'\rho \to_9 \lambda c^n c^k bW'\rho \to_{12} \lambda c^n c^k W'b\rho$

(13/14) If $fg > 0$,
$\lambda c^n(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g\rho$
$\to_{13} \lambda c^n(aab)^g(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})\rho$
$\to_{15} \lambda c^n \prod_{k=j}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g(ab)^f \prod_{k=1}^{j-1}((aab)^{p_k}(ab)^{q_k})\rho$
$\to_{14} \lambda c^n \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g(ab)^f\rho$
$\to_{15} \lambda c^n \prod_{k=j}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g(ab)^f \prod_{k=1}^{j-1}((aab)^{p_k}(ab)^{q_k})\rho.$

To show that Z is terminating (no infinite sequences of reductions), use the same weight-priority ordering as in Theorem 3.1, extended to include λ and ρ by making the weights of λ and ρ both 1, and letting the new priority list be $\lambda < c < c^{-1} < a < b < a^{-1} < b^{-1} < \rho$.

We still have to prove that the congruence relation generated by Z is the same as that generated by the relations Y of C_3. That is, we must show that each of (1)–(15) is a consequence of rules (A)–(H) and the following rule:

$$\lambda g^{\epsilon}Qg^{-\epsilon}\rho = \lambda Q\rho, \quad \epsilon = \pm 1,\ g \in \{a, b\} \tag{I}$$

where Q is any word in M_3. Firstly, note that we have already shown in Theorem 3.1 that T is equivalent to $S = \{(\mathrm{A}) \ldots (\mathrm{H})\}$. Thus, in establishing consequences of (A)–(I), we may use (1)–(10) of T, interpreted as relations. Secondly, observe that

$$\lambda c^n Wa\rho =_I \lambda ac^n Waa^{-1}\rho =_C \lambda ac^n W\rho =_{7,8} \lambda c^n aW\rho$$

$$\lambda c^n Wb\rho =_I \lambda bc^n Wbb^{-1}\rho =_E \lambda bc^n W\rho =_{9,10} \lambda c^n bW\rho$$

That is, a and b can be "pulled around" nonnegative words through powers of c. Since $c = aaa$, it can be similarly pulled around, and thus by repeated application of this process, any nonnegative word on a, b, c can be pulled around in this situation. This is recorded as the rule:

$$\lambda c^n XW\rho = \lambda c^n WX\rho, \tag{J}$$

for any nonnegative words X and W in M_3, and any integer n.

Now we can show that each reduction in Z is a consequence of Y. Since rules (1)–(10)

of Z constitute T, they are, by Theorem 3.1, consequences of S, a subset of Y. Each of the remaining rules is an instance of (J). In (11) a is moved to the front, in (12) b is moved to the back, in (13) $(aab)^g$ is moved to the front, in (14) $(ab)^f$ is moved to the back, and in (15) $\prod_{k=1}^{j-1}((aab)^{p_k}(ab)^{q_k})$ is moved to the back.

It remains to show that each relation in Y is in the congruence generated by Z. Observe that, as above, relations (A)–(H) are already done by Theorem 3.1, so the only relation to be considered is (I) and we may use (A)–(H) in demonstrating that it is a consequence of Z. Without loss of generality, we may take Q to be T-reduced,

$$Q = c^n L(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g R,$$

where $L \in \{e, b\}$ and $R \in \{e, a, aa\}$, $v + f + g \geq 0$ for otherwise we will first make it so, using relations in T, all of which are in Z.

First we look at the form (I_1)

$$\lambda a Q a^{-1}\rho = \lambda Q \rho.$$

In each case we will proceed from the left side to the right side. We will use $=_x$ to mean that we are substituting the right side of relation x for an instance of the left side of x, and $\overset{*}{=}_x$ for substituting the left side for the right side.

Case 1: $L = R = e$, $v = f = g = 0$, $(Q = c^n)$.

$$\lambda a c^n a^{-1}\rho =_{7,8} \lambda c^n a a^{-1}\rho =_C \lambda c^n \rho.$$

If $R = e$, the first steps are

$$\begin{aligned}\lambda a c^n L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a^{-1}\rho \\ =_1 \lambda a c^n L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g c^{-1} a a \rho \\ =_{7,8,10} \lambda c^{n-1} a L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a a \rho\end{aligned}$$

so we will start from this form.

Case 2: $L = R = e$, $f = 0$, $v = 0$, $g > 0$.

$$\begin{aligned}\lambda c^{n-1} a (aab)^g a a \rho &=_5 \lambda c^n b (aab)^{g-1} a a \rho \\ &=_{12} \lambda c^n (aab)^g \rho.\end{aligned}$$

Case 3: $L = R = e$, $f = 0$, $v > 0$.

$$\begin{aligned}\lambda c^{n-1} a \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a a \rho \\ =_5 \lambda c^n b (aab)^{p_1-1}(ab)^{q_1} \textstyle\prod_{k=2}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a a \rho \\ =_{12} \lambda c^n (aab)^{p_1-1}(ab)^{q_1} \textstyle\prod_{k=2}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a a b \rho \\ =_{13} \lambda c^n (aab)^{p_1+g}(ab)^{q_1} \textstyle\prod_{k=2}^{v}((aab)^{p_k}(ab)^{q_k})\rho \\ \overset{*}{=}_{13} \lambda c^n (aab)^{p_1}(ab)^{q_1} \textstyle\prod_{k=2}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g \rho.\end{aligned}$$

Case 4: $L = b$ or $f > 0$, $R = e$.

$$\begin{aligned}\lambda c^{n-1} a L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a a \rho \\ =_{11} \lambda c^{n-1} a a a L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g \rho \\ =_5 \lambda c^n L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g \rho.\end{aligned}$$

If $R = a$, the first steps are

$$\begin{aligned}\lambda a c^n L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a a^{-1}\rho \\ =_{7,8} \lambda c^n a L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a a^{-1}\rho \\ =_C \lambda c^n a L(ab)^f \textstyle\prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g \rho\end{aligned}$$

so we will start from this form.

Case 5: $L = e$, $R = a$.

$$\lambda c^n a(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g\rho \stackrel{*}{=}_{11} \lambda c^n(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a\rho.$$

Case 6: $L = b$, $R = a$.

$$\begin{aligned}\lambda c^n ab(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g\rho &=_{14} \lambda c^n \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g ab(ab)^f\rho\\ &\stackrel{*}{=}_{14} \lambda c^n(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g ab\rho\\ &\stackrel{*}{=}_{12} \lambda c^n b(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a\rho.\end{aligned}$$

If $R = aa$, the first steps are

$$\begin{aligned}\lambda ac^n L(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g aaa^{-1}\rho &=_C \lambda ac^n L(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a\rho\\ &=_{7,8} \lambda c^n aL(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a\rho\end{aligned}$$

so we will start from this form.

Case 7: $L = e$, $R = aa$.

$$\lambda c^n a(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a\rho \stackrel{*}{=}_{11} \lambda c^n(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g aa\rho.$$

Rule 11 is applicable (in reverse) in this case since $(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g aa$ is T-reduced, according to Corollary 3.2.

Case 8: $L = b$, $R = aa$.

$$\begin{aligned}\lambda c^n ab(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a\rho &=_{11} \lambda c^n aab(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g\rho\\ &=_{13} \lambda c^n(aab)^g aab(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})\rho\\ &\stackrel{*}{=}_{13} \lambda c^n(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g aab\rho\\ &\stackrel{*}{=}_{12} \lambda c^n b(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g aa\rho\end{aligned}$$

where the use of rule (11) in the first step is justified since $ab(ab)^f \prod_{k=1}^{v}((aab)^{p_k}(ab)^{q_k})(aab)^g a$ is T-reduced according to Corollary 3.2.

The form (I_2)

$$\lambda bQb^{-1}\rho = \lambda Q\rho$$

is much easier to resolve: again $Q = c^n W$, with W nonnegative and containing no c's, and W reduced under T.

$$\begin{aligned}\lambda bc^n W b^{-1}\rho &=_2 \lambda bc^n W c^{-1} b\rho =_{7,8,9,10} \lambda c^{n-1} bWb\rho\\ &=_{12} \lambda c^{n-1} Wbb\rho =_6 \lambda c^{n-1} Wc\rho =_{7,9} \lambda c^n W\rho.\end{aligned}$$

The forms $\lambda a^{-1}Qa\rho = \lambda Q\rho$ and $\lambda b^{-1}Qb\rho = \lambda Q\rho$ are now trivial to establish since

$$\lambda a^{-1}Qa\rho \stackrel{*}{=}_{I_1} \lambda aa^{-1}Qaa^{-1}\rho =_C \lambda Qaa^{-1}\rho =_C \lambda Q\rho;$$

$$\lambda b^{-1}Qb\rho \stackrel{*}{=}_{I_2} \lambda bb^{-1}Qbb^{-1}\rho =_E \lambda Qbb^{-1}\rho =_E \lambda Q\rho.$$

This completes the proof of Theorem 4.1.

COROLLARY 4.2. *Each braid on three strings is conjugate to exactly one of the following:*

(1) $c^n a$, n any integer,
(2) $c^n b$, n any integer,
(3) $c^n aa$, n any integer,
(4) $c^n (ab)^m$, n any integer, $m = 1, 2, 3, \ldots$,
(5) $c^n (aab)^m$, n any integer, $m = 1, 2, 3, \ldots$,
(6) $c^n \prod_{k=1}^{v} ((aab)^{p_k} (ab)^{q_k})$, n any integer, $v \geq 0$, with $p_1 q_1 \ldots p_v q_v$ its own smallest cycle.

PROOF. These are exactly the forms w such that $\lambda w \rho$ is irreducible under the rules in Z, as can be verified by inspection. QED.

By substituting back $a = \sigma_1 \sigma_2$, $b = \sigma_1 \sigma_2 \sigma_1$ and $c = \sigma_1 \sigma_2 \sigma_1 \sigma_2 \sigma_1 \sigma_2$, a complete set of conjugacy class representatives for B_3 in terms of σ_1 and σ_2 can be obtained.

5. Acknowledgements

The authors thank the referees for their careful readings of the paper which led to its improvement.

References

Artin, E. (1947). Theory of Braids. *Ann. Math.* **48**, 101–126.
Benninghofen, B., Kemmerich, S., Richter, M. (1987). Systems of reductions. Lecture Notes in Computer Science, vol. 277. Springer-Verlag, Berlin.
Birman, J. (1974). Braids, links and mapping class groups. *Studies in Math,* vol. **82**. Princeton University Press, Princeton.
Book, R. (1982). Confluent and other types of Thue systems. *J. Assoc. Comp. Mach.* **29**, 171–182.
Book, R. (1987). Thue systems as rewriting systems. *J. Symbolic Comp.* **3**, 39–68.
Dehn, M. (1911). Über unendliche diskontinuierliche Gruppen. *Math. Ann.* **71**, 116–144.
Elrifai, E., Morton, H. (1990). *Algorithms for Positive Braids.* Preprint.
Evans, T. (1951). On multiplicative systems defined by generators and relations. I. Normal form theorems. *Proc. Camb. Phil. Soc.* **47**, 637–649.
Freyd, P., Yetter, D., Hoste, J., Lickorish, W., Millett, K., Ocneanu, A. (1985). A new polynomial invariant of knots and links. *Bull. Amer. Math. Soc.* **12**, 239–246.
Garside, F. (1969). The braid group and other groups. *Quart. J. Math. Oxford (2)* **20**, 235–254.
Jacquemard, A. (1990). About the effective classification of conjugacy classes of braids. *J. Pure Appl. Alg.* **63**, 161–169.
Jones, V.F.R. (1985). A polynomial invariant for knots via von Neumann algebras. *Bull. Amer. Math. Soc.* **12**, 103–111.
Jones, V.F.R. (1987). Hecke algebra representations of braid groups and link polynomials. *Ann. of Math.* **126**, 335–388.
Kapur, D., Narendran, P. (1985). A finite Thue system with a decidable word problem and without equivalent finite canonical system. *Theoretical Comp. Sci.* **35**, 337–344.
Knuth, D., Bendix, P. (1970). Simple word problems in universal algebras. In: *Computational Problems in Abstract Algebra.* (J. Leech, ed.), pp. 263–297. New York: Pergamon Press.
Le Chenadec, P. (1986). Canonical forms in finitely presented algebras. Lecture Notes in Theoretical Computer Science, vol. 1. Pitman, London.
Markov, A.A. (1945). Foundations of the algebraic theory of braids. *Trudy mat. Inst. Steklov* **16** (in Russian).
Pedersen, J.F. (1985). The word problem in absorbing varieties. *Houston J. Math.* **11**, 575–590.
Thurston, W. (1992). Finite state algorithms for the braid group. In: *Word Processing in Groups.* (D. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, W. P. Thurston, eds), chapter 9. Wellesley, MA: A. K. Peters. Preprint.
Turaev, V.G. (1988). The Yang–Baxter equation and invariants of links. *Invent. Math.* **92**, 527–553.